IN THE CLAIMS

Please amend the claims as follows:

1 - 2. (canceled)

3. (currently amended) The method of claim 30, ~~1,~~ wherein the vulnerability includes ~~is~~ a vulnerability to a computer virus.

4. (currently amended) The method of claim 30, ~~2,~~ wherein the vulnerability includes ~~is~~ a vulnerability to computer hacking.

5. (currently amended) The method of claim 30 ~~1,~~ further comprising:

classifying the data processing systems storing replicas of the ~~first~~ resource as vulnerable, wherein the classifying is responsive to the predetermined number of changed replicas of the resource being exceeded and the predetermined time duration being exceeded.

6. (currently amended) The method of claim 30 ~~1,~~ further comprising:

replacing the replica of the ~~first~~ resource at each of the data processing systems ~~determined to be~~ storing a replica of the ~~first~~ resource, wherein the replacing is responsive to the predetermined number of changed replicas of the resource being exceeded and the predetermined time duration being exceeded.

7. (currently amended) The method of claim 30 ~~1,~~ further comprising:

patching the replica of the ~~first~~ resource at each of the data processing systems ~~determined to be~~ storing a replica of the ~~first~~ resource, wherein the patching is responsive to the predetermined number of changed replicas of the resource being exceeded and the predetermined time duration being exceeded.

8. (currently amended) The method of claim 7, further comprising:

prior to patching the replica of the ~~first~~ resource ~~with patch code~~, comparing a set of hash values representing all pre-requisite programs of ~~the~~ patch code with a ~~the~~ stored set of hash values ~~to identify matching hash codes~~; and

in response to identification of matching hash codes for all pre-requisite programs, determining that said patching of the replica of the ~~first~~ resource ~~with the patch code~~ should proceed.

9. (currently amended) The method of claim 30 ~~1,~~ the steps further comprising:

sending a notification of the vulnerability to each data processing system ~~determined to be~~ storing a replica of the ~~first~~ resource, wherein the sending is responsive to the predetermined number of changed replicas of the resource being exceeded and the predetermined time duration being exceeded.

10. (currently amended) The method of claim 9, further comprising:

~~responding to the determination of respective systems storing replicas of the first resource by~~ selecting a sequence of vulnerability-resolution instructions relevant to the vulnerability and including the selected instructions within the notification sent to each data processing system.

11 - 29. (canceled)

30. (new) A method within a network having a vulnerability, the method comprising the steps of:

computing first hash values derived from and representing a plurality of replicas of a resource, wherein the replicas of the resource are stored on respective data processing systems within a network;

storing the computed set of first hash values, wherein the storing includes:

storing identifications of the respective ones of said data processing systems

storing the replicas of the resource; and

storing time stamps for the hash values;

wherein the method further comprises:

computing at least second hash values for the replicas of the resource, wherein the computing of the at least second hash values is at a time after the computing of the first hash values;

computing current hash values for the replicas of the resource, wherein the computing of the current hash values is at a time after the computing of the at least second hash values;

comparing the hash values computed at successive times for each respective replica of the resource, in order to identify whether matching hash values exist;

comparing time stamps for matching hatch values of each respective replica of the resource;

computing, responsive to the time stamp comparison, a time duration during which the hash values of each respective replica of the resource remained unchanged;

detecting for a current time, responsive to the hash value comparison indicating that replicas of the resource have changed from one time to the current time, whether the changed replicas of the resource at the current time indicate a vulnerability, wherein the detecting comprises:

detecting whether the computed time duration prior to the current time exceeds a predetermined time duration; and

detecting whether changed replicas of the resource at the current time are more numerous than a predetermined number; and

wherein the method further comprises:

presenting a message for a user indicating the changed replicas of the resource are due to a vulnerability, wherein the presenting is responsive to the predetermined number of changed replicas of the resource being exceeded and the predetermined time duration being exceeded.

31. (new) An apparatus comprising:

a processor; and

a storage device connected to the processor, wherein the storage device has stored thereon a program, wherein the processor is operative to execute instructions of the program to implement a method comprising the steps of:

computing first hash values derived from and representing a plurality of replicas of a resource, wherein the replicas of the resource are stored on respective data processing systems within a network;

storing the computed set of first hash values, wherein the storing includes:

storing identifications of the respective ones of said data processing systems

storing the replicas of the resource; and

storing time stamps for the hash values;

wherein the method further comprises:

computing at least second hash values for the replicas of the resource, wherein the computing of the at least second hash values is at a time after the computing of the first hash values;

computing current hash values for the replicas of the resource, wherein the computing of the current hash values is at a time after the computing of the at least second hash values;

comparing the hash values computed at successive times for each respective replica of the resource, in order to identify whether matching hash values exist;

comparing time stamps for matching hatch values of each respective replica of the resource;

computing, responsive to the time stamp comparison, a time duration during which the hash values of each respective replica of the resource remained unchanged;

detecting for a current time, responsive to the hash value comparison indicating that replicas of the resource have changed from one time to the current time, whether the changed replicas of the resource at the current time indicate a vulnerability, wherein the detecting comprises:

detecting whether the computed time duration prior to the current time exceeds a predetermined time duration; and

detecting whether changed replicas of the resource at the current time are more numerous than a predetermined number; and

wherein the method further comprises:

presenting a message for a user indicating the changed replicas of the resource are due to a vulnerability, wherein the presenting is responsive to the predetermined number of changed replicas of the resource being exceeded and the predetermined time duration being exceeded.

32. (new) The apparatus of claim 31, wherein the vulnerability includes a vulnerability to a computer virus.

33. (new) The apparatus of claim 31, wherein the vulnerability includes a vulnerability to computer hacking.

34. (new) The apparatus of claim 31, the steps further comprising:

classifying the data processing systems storing replicas of the resource as vulnerable, wherein the classifying is responsive to the predetermined number of changed replicas of the resource being exceeded and the predetermined time duration being exceeded.

35. (new) The apparatus of claim 31, the steps further comprising:

replacing the replica of the resource at each of the data processing systems storing a replica of the resource, wherein the replacing is responsive to the predetermined number of changed replicas of the resource being exceeded and the predetermined time duration being exceeded.

36. (new) The apparatus of claim 31, the steps further comprising:

patching the replica of the resource at each of the data processing systems storing a replica of the resource, wherein the patching is responsive to the predetermined number of changed replicas of the resource being exceeded and the predetermined time duration being exceeded.

37. (new) The apparatus of claim 36, the steps further comprising:

prior to patching the replica of the resource, comparing a set of hash values representing all pre-requisite programs of patch code with a stored set of hash values; and

in response to identification of matching hash codes for all pre-requisite programs, determining that said patching of the replica of the resource should proceed.

38. (new) The apparatus of claim 31, the steps further comprising:

sending a notification of the vulnerability to each data processing system storing a replica of the resource, wherein the sending is responsive to the predetermined number of changed replicas of the resource being exceeded and the predetermined time duration being exceeded.

39. (new) The apparatus of claim 38, the steps further comprising:

selecting a sequence of vulnerability-resolution instructions relevant to the vulnerability and including the selected instructions within the notification sent to each data processing system.

40. (new) A computer program product, stored on a tangible, computer readable medium, said computer program product having instructions for execution by a computer system, wherein the instructions, when executed by the computer system, cause the computer system to implement a method comprising the steps of:

computing first hash values derived from and representing a plurality of replicas of a resource, wherein the replicas of the resource are stored on respective data processing systems within a network;

storing the computed set of first hash values, wherein the storing includes:

storing identifications of the respective ones of said data processing systems storing the replicas of the resource; and

storing time stamps for the hash values;

wherein the method further comprises:

computing at least second hash values for the replicas of the resource, wherein the computing of the at least second hash values is at a time after the computing of the first hash values;

computing current hash values for the replicas of the resource, wherein the computing of the current hash values is at a time after the computing of the at least second hash values;

comparing the hash values computed at successive times for each respective replica of the resource, in order to identify whether matching hash values exist;

comparing time stamps for matching hatch values of each respective replica of the resource;

computing, responsive to the time stamp comparison, a time duration during which the hash values of each respective replica of the resource remained unchanged;

detecting for a current time, responsive to the hash value comparison indicating that replicas of the resource have changed from one time to the current time, whether the changed replicas of the resource at the current time indicate a vulnerability, wherein the detecting comprises:

detecting whether the computed time duration prior to the current time exceeds a predetermined time duration; and

detecting whether changed replicas of the resource at the current time are more numerous than a predetermined number; and

wherein the method further comprises:

presenting a message for a user indicating the changed replicas of the resource are due to a vulnerability, wherein the presenting is responsive to the predetermined number of changed replicas of the resource being exceeded and the predetermined time duration being exceeded.

41. (new) The computer program product of claim 40, wherein the vulnerability includes a vulnerability to a computer virus.

42. (new) The computer program product of claim 40, wherein the vulnerability includes a vulnerability to computer hacking.

43. (new) The computer program product of claim 40, the steps further comprising:

classifying the data processing systems storing replicas of the resource as vulnerable, wherein the classifying is responsive to the predetermined number of changed replicas of the resource being exceeded and the predetermined time duration being exceeded.

44. (new) The computer program product of claim 40, the steps further comprising:

replacing the replica of the resource at each of the data processing systems storing a replica of the resource, wherein the replacing is responsive to the predetermined number of changed replicas of the resource being exceeded and the predetermined time duration being exceeded.

45. (new) The computer program product of claim 40, the steps further comprising:

patching the replica of the resource at each of the data processing systems storing a replica of the resource, wherein the patching is responsive to the predetermined number of changed replicas of the resource being exceeded and the predetermined time duration being exceeded.

46. (new) The computer program product of claim 45, the steps further comprising:

prior to patching the replica of the resource, comparing a set of hash values representing all pre-requisite programs of patch code with a stored set of hash values; and

in response to identification of matching hash codes for all pre-requisite programs, determining that said patching of the replica of the resource should proceed.

47. (new) The computer program product of claim 40, the steps further comprising:

sending a notification of the vulnerability to each data processing system storing a replica of the resource, wherein the sending is responsive to the predetermined number of changed replicas of the resource being exceeded and the predetermined time duration being exceeded.

48. (new) The computer program product of claim 47, the steps further comprising:

selecting a sequence of vulnerability-resolution instructions relevant to the vulnerability and including the selected instructions within the notification sent to each data processing system.